

FIG. 2

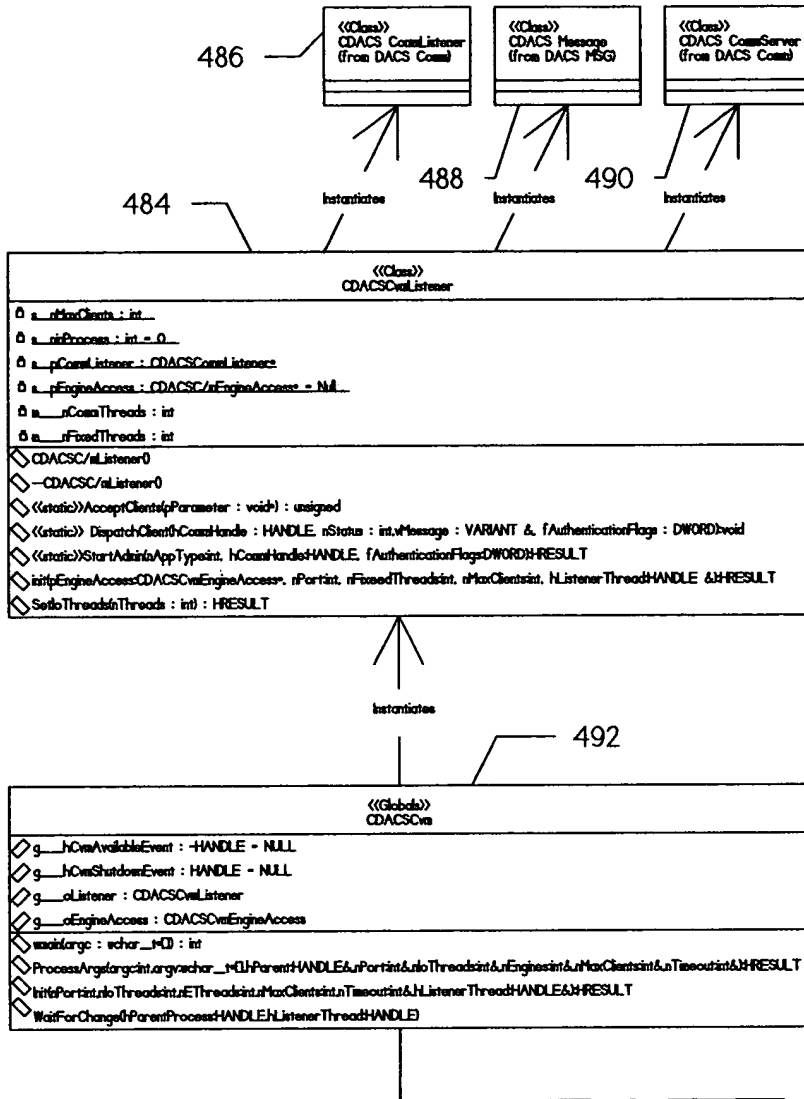


FIG. 3A



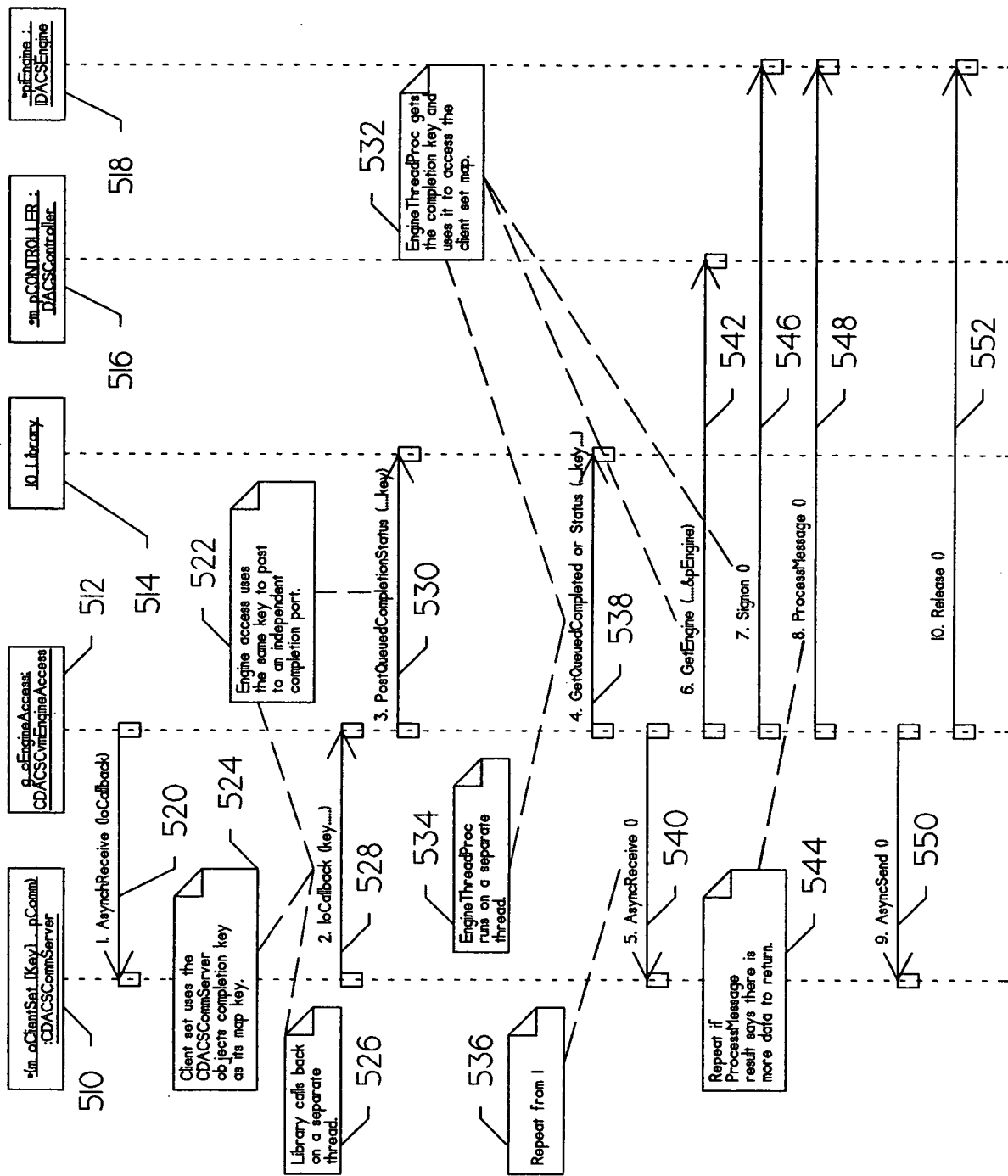


FIG. 4

Message #	Description	Fig. 16 Reference
1	Ask to receive a message from the client. Caller is AddClient or EngineThreadProc.	520
2	DACSComm library calls loCallback when a complete message has been received on the socket associated with this object. Multiple such calls can happen simultaneously on parallel threads for different clients.	528
3	loCallback posts the incoming message to the engine access queue, and returns to the DACSComm library.	530
4	EngineThreadProc picks the oldest message off the queue and processes it. Multiple instances of EngineThreadProc can be doing this simultaneously on parallel threads. The client set structure for the given key provides a pointer to the CDACSCommServer object, and the information for calling the engine.	538
5	EngineThreadProc calls AsyncReceive before it calls the engine. This gives the client an opportunity to send a "Cancel" message.	540
6	Get an engine interface from the DACS Controller, using the instance name stored in the client set for this key.	542
7	Use credentials stored in the client set entry to sign onto the engine.	546
8	Marshal the client's message to the engine interface. Receive a response message in the same parameter.	548
9	Send the response back to the client. Do not ask for a callback.	550
10	Release the engine interface when there is no more data to receive from the engine and send to the client.	552

FIG. 5